

INTERNET RELAY CHAT

Dan Bruce, David Clark and Héctor D. Menéndez

Department of Computer Science
University College London

November 27, 2017

Index

- 1** Introduction
- 2** IRC Specifications
- 3** IRC Concepts
- 4** Messages
- 5** Exercise

Introduction

IRC is a TCP/IP protocol.

It is implemented as a client-server or client-servers approach.

Servers

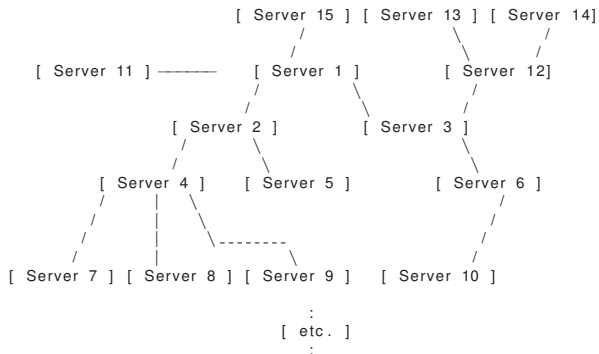
They are the main IRC structure.

They provide an access point for the clients allowing them to chat. The set of servers form the IRC Network.

The configuration is an extended tree.

Each server is a central node for its neighbours.

Servers



Clients

A client is something connected to the Server, but another server.

It has a unique nick with, at most, 9 characters.

Every server needs the following information about all the clients:

- Real host name associated to the client.
- Username of the user in the host.
- Client's server.

Operators

Special clients with some privileges in the network.

They have to be able to do basic task to prevent problems in the network.

The can also eliminate a user.

Channels

It is a group.

Created by the first client joining the channel. When the channel has no clients, it disappears.

The name is a string with no white space (' ') or commas (',').

There are two types of channels:

- Distributed channel, known by all the network servers. It is denoted by '#'.
- Local channel for the current server. It is denoted by '&'.

Channels

In order to create a new channel, a user must JOIN the channel.

If the channel exists, the user is connected to the channel. Otherwise, he creates the channel and becomes an operator for that channel.

A user can belong to different channels at the same time. The recommended limit is 10.

Channel Operators

He is th channel “owner”.

They have some “privileges” to keep the channel control.

They are identified by '@’.

Channel Operators

They use the following commands:

- KICK - throws out a user
- MODE - changes channel mode
- INVITE - invites a user to join the channel
- TOPIC - changes channel topic with mode +t

Discussion

Two types of connexions:

- Server-server.
- Client-server.

Messages

Servers and Clients send messages to each other.

When the message has a valid command, the other part waits for an answer.

The communication is asynchronous.

Each message has three main parts (separated by white spaces):

- Prefix (optional, denoted by ':').
- Command.
- Parameters.

Messages: The prefix

Used by the servers to indicate the real origin of a message.

Client might not use it.

The only prefix used by clients is the one that contain their nicks.

Messages

Every IRC message finishes with CR-LF.

The limit is 512 characters, including CR-LF.

Message Format: 'pseudo' BNF

CR and LF are message separators.

Empty messages are ignored.

The message is divided in <prefix>, <command> and parameter list, formed by <intermediate parameter> or <final parameter>.

Message Format: 'pseudo' BNF

```
<message> ::= [ ':' <prefix> <SPACE> ] <command> <parameter> <CrLf>
<prefix> ::= <server name> | <nick> [ '!' <user> ] [ '@' <host> ]
<command> ::= <letter> { <letter> } | <number> <number> <number>
<SPACE> ::= ' ' { ' ' }
<parameter> ::= <SPACE> [ ':' <final parameter> |
                        <intermediate parameter> <parameter> ]

<intermediate parameter> ::= <Non-empty octet sequence without
                            SPACE, NUL, CR o LF, the first is not ':'>
<final parameter> ::= <Any sequence, sometimes empty, without
                        NUL, CR o LF>

<CrLf> ::= CR LF
```

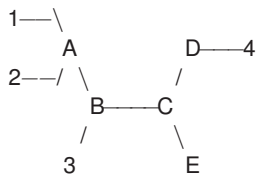
Numerical Answers

Usually, messages sent to the server generate an answer.

The most frequent answer is a numeric answer for both errors and normal answers.

IRC Concepts

Let's see the following tree:



Servers: A, B, C, D, E

Clients: 1, 2, 3, 4

One-to-one Communication

Between clients.

The path is the shortest one.

One-to-Many Communication

Forum between different users in a specific channel.

To a List

Inefficient.

The client provides a list of targets and the server distributes the message generating one copy per target.

To a group (channel)

Dynamic

The conversation is submit to those servers that have users in the specific channel.

To a server

Use for the Operators to send messages to a lot of related users.

It is send to users related to a specific host.

One-to-All

Broadcast to all servers and clients.

Client-to-client

It does not exist

Client-to-Server

Almost all messages.

Server-to-Server

Almost all messages generated by a server are submitted to all connected servers.

Messages

Format:

```
:Name COMMAND parameter list
```

Connexion Register

- 1 Password Message
- 2 Nick Message
- 3 User Message

Password Message

Command: PASS

Parameters: <password>

Description: establish a connexion password

Example: PASS passhere

Nick Message

Command: NICK

Parameters: <nick> [<jumpercounter>]

Description: set a nick to a user

Example: NICK Wiz ; New nick "Wiz".

User Message

Command: USER

Parameters: <user name> <host name> <server name> <real name>

Description: at the beginning of the connection to indicate: user info.

Example: `USER guest tolmoon tolsun :Ronnie Reagan ;` User name "guest" real name: "Ronnie Reagan".

Server Message

Command: SERVER

Parameters: <server name> <jumps> <info>

Description: indicates to a server that the other sender is a server

Example: `SERVER test oulu.fi 1 :[tolsun oulu.fi] experimental server ;` The new server test oulu.fi is introduced and register.

Oper

Command: OPER

Parameters: <user> <password>

Description: a normal user obtains Operator privileges

Example: OPER foo bar ; The user register itself with username “foo” and password “bar”

Quit

Command: QUIT

Parameters: [<Message>]

Description: The client leaves with a message

Example: QUIT :bye ; Message Format

Server Quit

Command: SQUIT

Parameters: <server> <comment>

Description: Close the connection between server.

Example: SQUIT tolsun.oulu.fi :Closing link ;The link with tolsun.oulu.fi has finished.

Messages

Channel manipulation according to its properties (channel mode) and content (clients)

JOIN Message

Command: JOIN

Parameters: <channel>{,<channel>} [<password>{,<password>}]

Description: to listen in a specific channel.

Example: JOIN # foobar ; join channel # foobar.

Leave a channel

Command: PART

Parameters: [<Channel>]

Description: The user leaves the specific channel.

Example: PART # twilightzone ; leave channel "# twilightzone"

Mode Message

Command: MODE

Parameters: <channel> { [+| -]| o| p| s| i| t| n| b| v } [<limit>] [<user>]
[<mask>] | <nick> { [+| -]| i| w| s| so }

Description: Changes modes from users and channels.

Example1: MODE # Finnish +im ; Channel # Finnish is moderated and only for invited users

Example2: :Angel MODE Angel +i ; Angel messages are invisible

Topic Message

Command: TOPIC

Parameters: <channel> [<topic>]

Description: changes or shows the channel topic.

Example: `:Wiz TOPIC # test :New Topic`; User WiZ changes the topic of a channel.

Name Messages

Command: NAMES

Parameters: [`<channel>`{,`<channel>`}]

Description: list all names from the channels (which are visible).

Example: NAMES # twilightzone,# 42 ;list visible users of # 42 and # twilightzone

Channel List

Command: LIST

Parameters: [<channel>{,<channel>} [<server>]]

Description: list channels and topics.

Example: LIST # twilightzone, # 42 ;list channels # twilightzone y
42

Mensaje de invitación a un canal

Command: INVITE

Parameters: <nick> <channel>

Description: invite users to a channel.

Example: :Angel INVITE Wiz # Dust ;Angel invites WiZ to channel # Dust

Temporal Banner

Command: KICK

Parameters: <channel> <user> [<commend>]

Description: eliminates a user from the channel

Example: KICK & Melbourne Matthew ; Bans Matthew from & Melbourne

Sending Messages

Client communication.

PRIVMSG and NOTICE are the only available commands.

Private Messages

Command: PRIVMSG

Parameters: <receptor>{,<receptor>} <text>

Description: send public or private messages among users.

Examples:

- :Angel PRIVMSG Wiz :Hello ; Message from Angel to Wiz.
- PRIVMSG Angel :Hi :) ;Message for Angel.
- PRIVMSG # Room :Hey ;Message from channel #Room.
- PRIVMSG jto@tolsun oulu.fi :Hello ; Message to server tolsun oulu.fi an user "jto"
- PRIVMSG \$*.fi :Server tolsun oulu.fi rebooted ; message to all the users from a server *.fi

Notice Messages

Command: NOTICE

Parameters: <nick> <texto>

Description: similar to PRIVMSG. You can not send automatic answers to a NOTICE

Other Messages

With no Category

“PING”

Command: PING

Parameters: <server1> [<server2>]

Description: check the state of another server or client

Example: PING WiZ ;PING to nick WiZ

“PONG”

Command: PONG

Parameters: <daemon> [<daemon>]

Description: answer to PING

Example: PONG csd.bu.edu tolsun.oulu.fi ;message PONG from csd.bu.edu to tolsun.oulu.fi

Error

Command: ERROR

Parameters: <error message>

Description: ERROR is to notify problems to the Operators

Example: ERROR :server *.fi not longer exists

Exercise

Use Wireshark to describe the packet exchange of the pcap file in Moodle.